Abstract

A method and apparatus for robustly enhanced Class of Service (COS) at the application layer permits highly flexible privilege based access and enables implementation of complex policies and rules for classification and differentiation of services. Differentiation facilitates categorization of traffic to permit flexible design and implementation of multiple Class of Service levels. A routing host is configured to receive all client requests for sites and virtual sites implemented on a plurality of service hosts or back-end servers. A monitoring processor incorporating an Adaptive Policy Engine, in communication with the router (and agents on back-end servers) dynamically monitors workload and availability of servers to enable requests to be sent to the most appropriate and optimal server. Incoming traffic is first processed to assign a class. The APE is employed to monitor the incoming traffic to the routing host. Traffic is measured to each hosted site and further, to each class of a hosted site. The APE has a rules based engine that correlates this information and uses it to come up with a dynamic, real time balancing scheme for each hosted site. The APE or policy engine in conjunction with the router then intelligently distributes incoming traffic to the most available and/or efficient server within each class or "cluster," by using one or more of a plurality of selectable load distribution algorithms, so that service level commitments are met. Intelligent agents deployed on each of the back-end servers monitor several server attributes/parameters and report back to the policy engine at the router. Class of service (COS) involves the classification of incoming requests by the policy engine. Backend server sites are clustered into virtual user definable cluster groups. Each cluster group can be managed/designated with a particular class of service. Based on information/parameters received, the composition of the clusters can be changed dynamically so that SLA parameters or metrics are within an acceptable range. Based on its class, the connection/request will be directed to one of the clusters. The specific machine selected will depend upon the load balancing algorithm defined for the cluster or class, and implemented as a function of the parameters reported to the policy engine, for making load balancing decisions.